Lista de exercícios

Introdução (Capítulo 1)

1) Faça todos os "Exercícios Propostos" do livro texto (Linguagem C, Luís Damas), página 17.

Tipos de dados básicos (Capítulo 2)

2) Implemente e estude o "prog0217.c" do livro texto, página 46.

3) Implemente e estude o "prog0218.c" do livro texto, página 46.

4) Implemente e estude o "prog0219.c" do livro texto, página 47.

5) Faça todos os "Exercícios Propostos" do livro texto, página 47.

6) Faça um programa para calcular o número de vértices de um cubo com 6 faces e 12 arestas. A relação entre vértices, arestas e faces de um objeto geométrico é dada por:
"vértices + faces = arestas + 2".

7) Sabe-se que o valor de cada 1000 litros de água corresponde a 2% do salário mínimo. Faça um programa para receber o valor do salário mínimo e a quantidade de água consumida em uma residência por mês. O algoritmo deverá calcular e mostrar: a) o valor da conta de água. b) o valor a ser pago com desconto de 15%.

8) Faça um programa que recebe dois valores na variáveis A e B e, em seguida, troca o conteúdo destas variáveis. Por exemplo, se o usuário digitar A=5 e B=3, o programa deverá trocar os valores de tal maneira que A seja igual a 3 e B igual a 5.

9) Num triângulo retângulo, segundo Pitágoras, o quadrado da hipotenusa (a) é igual a soma dos quadrados dos catetos (b e c). Faça um algoritmo que recebe o valor dos catetos e imprime o valor da hipotenusa.

10) Escreva um programa para determinar a quantidade de litros de combustível gastos em uma viagem por um automóvel que faz 12 km/litro. Para isso, sabe-se que o tempo gasto na viagem é T=35 min e a velocidade média do automóvel é V = 80 km/h.

11) Faça um programa que calcula a média de quatro números introduzidos pelo usuário.

12) Faça um programa que leia um número inteiro de 4 dígitos e escreva-o invertido. Por exemplo, se o número lido for 2548, o resultado será 8452. Dica: utilize o comando "%" que retorna resto da divisão entre 2 números inteiros.

Testes e condições (Capítulo 3)

13) Faça os "Exercícios Propostos" 13 do livro texto, página 78.

14) Faça os "Exercícios Propostos" 14 do livro texto, página 78.

15) Resolva as seguintes expressões lógicas:

a) não (V e (V ou F))

b) não (V e não (V ou F))

c) (F ou V) e F

16) Faça um programa que leia a idade de uma pessoa e diga-lhe se é maior de idade ou não (idade >= 18) .

17) Faça um programa que leia dois números inteiros e determine qual dos dois é maior. Considere que os dois números serão diferentes.

18) Faça um programa que leia dois números inteiros e determine qual dos dois é maior. Considere que os dois números podem ser iguais. Neste caso, o programa deve escrever uma mensagem para o usuário informando-o de que deve entrar com números diferentes.

19) Faça um programa que leia 3 números e determine quantos são iguais.

20) Faça um programa capaz de identificar se um número é igual a 1, 5 ou 10. Caso não seja nenhum desses valores, retornar a mensagem "Valor inválido".

21) Faça um programa capaz de identificar se um número é par ou ímpar.

22) Faça um programa capaz de identificar se um número é positivo, negativo ou zero.

23) Faça um programa capaz de identificar se um número é um ano bissexto. Considere que para o ano ser bissexto basta que seja divisível por 400. Caso contrário, precisa ser divisível por 4 e não ser divisível por 100. Faça uma condição composta que englobe todas as regras para a definição do ano bissexto.

24) Faça um algoritmo que simule uma calculadora com as quarto operações básicas (+,-,*,/). O algoritmo deve solicitar ao usuário a entrada de dois operandos e da operação a ser executada, na forma de menu. Dependendo da opção escolhida, deve ser executada a operação solicitada e escrito seu resultado. Utilize uma variável do tipo caractere para armazenar a operação e utilize o comando caso para escolher a operação a partir do operador. (Solução PROG0316.c, página 73 do livro texto).

Estrutura de repetição - Laços (Capítulo 4)

25) Faça um programa que leia 10 números digitados pelo usuário e retorne a soma e a média desses valores.

26) Altere o programa anterior, de tal maneira que o usuário informe a quantidade de números que serão digitados (ou seja, o valor "10" não deve ser fixo no programa).

27) Faça um programa que calcule a multiplicação de 2 números inteiros sem utilizar o operador "*". Em vez disso, utilize apenas o operador de adição "+".

28) Faça um programa que calcule o fatorial de um número. Se o número for menor do que zero, então o algoritmo deverá informar ao usuário que o valor é inválido.

29) Faça um programa que solicita ao usuário dois números inteiros diferentes "n1" e "n2" e calcula a soma de todos os números ímpares dentro do intervalo definidor por [n1,...,n2]. Considere que n1 é sempre menor do que n2.

30) Alltere o programa anterior de tal maneira que quando o usuário digitar um intervalo inválido (n1>n2), o programa irá solicitar novos valores para n1 e n2.

31) Faça um programa que leia um número inteiro positivo e determine se este é primo ou não. Por definição, um número é primo quando é divisível somente por si próprio e por 1.

32) Otimize o programa anterior com base nas seguintes considerações:

- Números pares (com exceção do 2) não podem ser primos, visto que são divisíveis por 2. Se um número não for divisível por 2, não será divisível por nenhum outro número par. Portanto, com excecão do número 2, somente é necessário testar números ímpares.

- É mais fácil que um número seja divisível por um número pequeno do que por um número maior. Portanto, se iniciarmos a procura do divisor de baixo para cima, ao invés de cima para baixo teremos chance de encontrar o número muito antes.

- Nenhum número pode ser divisível por outro número maior que a metade dele. Portanto, não precisamos testar a divisibilidade dos números na faixa entre a metade e o próprio número.

Dado um valor de E fornecido pelo usuário, calcular S=1 + 1/2 + 1/4 +1/6 + … até que um termo da série seja menor do que E.

33) Faça um programa para calcular o valor de S, dado por:

S= 1/1 + 3/2 + 5/3 + 7/4 + … + 99/50

34) Faça um programa para calcular o valor de S, dado por:

S= 1/1 – 2/4 + 3/9 – 4/16 + … - 10/100

35) Faça um programa para calcular e mostrar o valor de PI, usando a série:

PI=4- 4/3 + 4/5 – 4/7 + 4/9 -... até que um termo seja menor do que 0.0001, em valor absoluto.

36) Faça um programa usando o comando "for" para calcular o seguinte somatório:

$$\sum_{i=3}^{n}(5*i+2)$$

em que "n" é definido pelo usuário.

37) Os números naturais menores do que 10 e múltiplos de 3 ou 5 são: 3, 5, 6 e 9. A soma destes múltiplos é 23. Faça um programa que encontre a soma de todos os múltiplos de 3 ou 5 menores do que 1000.

Resposta: 233168

38) Cada novo termo da sequência de Fibonacci é gerado pela adição dos 2 termos anteriores. Ao iniciar a sequência com 1 e 2, os dez primeiros termos são: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, … Faça um programa que encontre a soma dos números pares da sequência de Fibonacci cujo termo não exceda 4 milhões.

Resposta: 4613732

39) O número 2520 é divisível (resto zero) por todos números inteiros de 1 a 10. Faça um programa que encontre o menor número inteiro positivo divisível por todos os inteiros de 1 a 20.

Resposta: 232792560

40) Escreva um programa que solicita ao usuário um valor entre 3 e 18. Este valor representa a **soma** dos valores do lançamento de 3 dados. Em seguida, o programa deve imprimir todas as possibilidades de que no lançamento dos 3 dados a soma de seus valores seja igual ao valor informado pelo usuário. Por exemplo, caso o usuário digite o valor 10, o programa deverá exibir: 6,3,1; 1,6,3; 6,2,2; 5,3,2 etc. Você pode exibir sequências iguais em dados diferentes, por exemplo, as sequências 6,3,1 e 1,6,3 possuem os mesmos números, mas em dados diferentes.

Funcões e Procedimentos (Capítulo 5)

41) Estude o Exercício de Exame que contém as funções “tiro” e “liro” na página 122 do livro texto (Luís Damas).
42) Faça o exercício proposto número 1.
43) Faça o exercício proposto número 2.
44) Faça o exercício proposto número 3.
45) Faça o exercício proposto número 4.
46) Faça o exercício proposto número 5.
47) Faça o exercício proposto número 6.
48) Faça o exercício proposto número 7.
49) Faça o exercício proposto número 8.
50) Faça o exercício proposto número 9.
51) Faça o exercício proposto número 10.
52) Faça o exercício proposto número 11.
53) Faça o exercício proposto número 12.
54) Faça o exercício proposto número 13.
55) Faça o exercício proposto número 14.
56) Faça o exercício proposto número 15.
57) Faça o exercício proposto número 16.

58) O “máximo divisor comum” de dois números inteiros A e B pode ser escrito da seguinte maneira:

```
enquanto B for diferente de zero
inicio
  r = resto da divisão de A por B;
  A = B;
  B = r;
fim
mdc = A;
```

Faça uma função em C que recebe dois valores A e B e retorna o MDC entre eles. Implemente o programa principal para testar a função implementada.

59) Considere a seguinte definição de ano bissexto (ano em que o mês de fevereiro tem 29 dias).

Um ano não divisível por 100 e divisível por 4 é bissexto;
Um ano divisível por 100 e divisível por 400 é bissexto;
Os demais anos não são bissextos.

Escreva uma função que recebe como parâmetros 3 números inteiros correspondendo aos valores de dia, mês e ano, respectivamente, e retorna o número de dias já transcorridos neste ano. Teste o programa para diversas datas.

60) Faça um programa para identificar se um número inteiro positivo é primo. Você deve implementar uma funcão específica que recebe um número inteiro e retorna o valor “0” caso o número NÃO seja primo e “1” caso contrário.

61) Faça um programa para calculcar a soma dos N primeiros números primos, sendo N definido pelo usuário na função principal “main()”. O programa deverá ter as funções “Soma_Primos” e “Primo”, sendo que a primeira será responsável pela soma dos números que forem primos e a segunda será responsável por verificar se o número em questão é primo ou não.

Vetores (Capítulo 6)

62) Escreva um programa para armazenar 15 números inteiros em um vetor. Os números deverão ser gerados aleatoriamente. O vetor deverá ser uma variável local dentro da função "main". Em seguida, implemente funções que recebem o vetor como parâmetro e retornam:

    1. a quantidade de números pares do vetor;
    2. a soma dos números ímpares do vetor;
    3. a quantidade de números com valor maior do que a média dos números do vetor;
    4. o maior valor do vetor;
    5. a maior diferença em valor absoluto entre os elementos consecutivos do vetor;

Implemente um procedimento para:

    6. exibir o 3o. maior elemento do vetor. Por exemplo: v={2,4,4,1,3,6,5,6} → 3o. Maior= 4.
    7. ordenar o vetor. No programa principal, exiba o vetor antes e depois de ordená-lo.

    8. eliminar números repetidos do vetor. Os números repetidos devem ser substituídos por novos números. Este procedimento deverá exibir na tela todos os elementos do vetor, antes e depois de eliminar os números repetidos.

    9. Após implementar todas a funções e procedimentos acima, você deverá implementar um menu onde o usuário poderá escolher uma das opções acima (1—8). Adicione a opção "9" para sair do programa.

63) Faça o exercício proposto número 2 – capítulo 6 (página 151).

64) Faça o exercício proposto número 5 – capítulo 6 (página 152).

65) Faça o exercício proposto número 6 – capítulo 6 (página 152).

66) Faça o exercício proposto número 7 – capítulo 6 (página 152).

Matrizes (Capítulo 6)

67) Escreva um programa para ler duas matrizes (3x3) e gravar a soma das duas em uma terceira matriz.

68) Escreva um programa que armazena 16 valores em uma matriz 4x4. Os números deverão ser gerados aleatoriamente. A matriz deverá ser uma variável local dentro da função "main". Em seguinda, implemente funções que recebem a matriz como parâmetro e retornam:

    1. a quantidade de números pares da matriz;

    2. a soma dos números ímpares da matriz;

    3. a quantidade de números com valor maior do que a média dos números da matriz;

    4. o maior valor da matriz;

    5. o segundo maior valor da matriz;

Implemente um procedimento para:

    6. eliminar os números repetidos da matriz. Os números repetidos devem ser substituídos por novos números. Este procedimento deverá exibir na tela todos os elementos da matriz, antes e depois de eliminar os números repetidos.

7. Após implementar todas a funções e procedimentos acima, você deverá implementar um menu onde o usuário poderá escolher uma das opções acima (1—6). Adicione a opção "7" para sair do programa.

69) Escreva um programa que carrega uma matriz de 12x4 com os valores das vendas de uma loja (estes valores podem ser gerados aleatoriamente). Cada linha da matriz representa um mês do ano e cada coluna, uma semana do mês. Os dados devem ser processados de forma a produzir as seguintes informações:

a) Total vendido em cada mês do ano;
b) A soma de todas as vendas da primeira semana de cada mês;
c) Total vendido no ano.

Implemente uma função para cada um dos itens acima.

70) Uma empresa tem registrado em uma tabela os consumos mensais de energia elétrica dos anos de 2003 até 2010. Cada linha representa um ano e cada coluna representa um mês. Considerando esses dados, faça um algoritmo para processar a tabela e produzir as seguintes informações:
1. Consumo médio em cada um dos meses;
   Mês/ano em que houve o maior gasto com energia.

71)
Implemente um programa que preenche com valores aleatórios uma matriz MxN e:

1. armazena em um vetor *v1* o **maior** elemento cadastrado em cada coluna da matriz;
2. armazena em um vetor *v2* o **menor** elemento cadastrado em cada coluna da matriz;

Ao final, o programa deve exibir: a matriz original, o vetor dos maiores elementos e o vetor dos menores elementos.

Obs.:
- Os valores de M e N devem ser definidos por meio da diretiva "#define"
- Você pode utilizar no máximo dois comandos "for" para encontrar os menores e maiores elementos das colunas.
- A solução deve ser implementada no programa principal ("main"), sem utilizar funções.

72) Uma fábrica produz dois tipos de motores M1 e M2. Nos meses de janeiro a dezembro o número de motores produzidos foi registrado na tabela 1. O setor de controle de vendas tem uma tabela de custos e de lucro (em mil reais) obtidos com cada motor (conforme a tabela 2). Escreva um programa preenche as tabelas 1 e 2 com números aleatórios e, em seguida calcula: a) o lucro em cada um dos meses e, b) o lucro anual total.

	Tabela 1	
Meses	**M1**	**M2**
Jan	30	20
Fev	5	10
...	...	...
Dez	18	28

Tabela 2

Motor	**Custo**	**Lucro**
M1	10	3
M2	15	2

Strings (Capítulo 7)

73) Implemente um procedimento que coloca em ordem crescente os caracteres de um string "s" passado como parâmetro.

Implemente as seguintes funções:

74) **"char * strchar (char *s, char ch)"** que retorna o endereço da primeira ocorrência de "ch" em "s"; caso não exista, retorna NULL (NULL é uma constante simbólica que indica que o ponteiro não aponta para nenhuma variável). Implemente a função principal ("main") para testar a função strchar.

75) **"char * strchar (char *s, char ch)"** que retorna o endereço da última ocorrência de "ch" em "s"; caso não exista, retorna NULL (NULL é uma constante simbólica que indica que o ponteiro não aponta para nenhuma variável). Implemente a função principal ("main") para testar a função strchar.

76) **"void imprime_sobrenome (char *nome_completo)"** que recebe o nome completo de uma pessoa e imprime o último nome. Por exemplo, para o nome completo "Jose Maria da Silva", o programa deve exibir: "Silva".

77) **"char *strstr (char *str1, char *str2)"** que retorna o endereço de str1 em que ocorre pela primeira vez a substring str2. Caso não exista, retorna NULL.

78) **"void calc(int * v, int num, int * xmin, int * xmax)"** que coloca nas posições de memórias apontadas por "xmin" e "xmax" o menor e o maior valor (respectivamente) existentes nos "num" primeiros inteiros do vetor "v".

79) **"void eliminar(char * v, char ch)"** que elimina todas as ocorrências do caracter "ch" no vetor de caracteres "v". Ao eliminar um caracter, os outros elementos do vetor devem ser movidos para esquerda. Por exemplo: *v = "programacao de computadores"* e *ch = 'a'*. Ao final do procedimento o conteúdo de "v" deverá ser "progrmco de computdores". Você não pode utilizar um vetor auxiliar para implementar o procedimento.

Ponteiros e passagem de parâmetros (Capítulo 8 e 9)

80) Faça o exercício proposto número 1 – capítulo 8 (página 200).
81) Faça o exercício proposto número 2 – capítulo 8 (página 200).
82) Faça o exercício proposto número 6 – capítulo 8 (página 200).
83) Faça o exercício proposto número 8 – capítulo 8 (página 201).
84) Faça o exercício proposto número 9 – capítulo 8 (página 201).
85) Faça o exercício proposto número 10 – capítulo 8 (página 201).

Arquivos (Capítulo 10)

86) Implemente um programa que conta quantas vezes a letra "a" aparece em um arquivo texto.

87) Implemente um programa que calcula e apresenta a soma duas matrizes de dimensão 5x5. Estas matrizes devem estar salvas em um único arquivo binário. Observe que o arquivo binário que contém as matrizes, também deve ser gerado pelo programa.

88) Implemente uma agenda de contatos que armazana "nome", "idade" e "numero de telefone". A agenda deve ser gravada em arquivo binário. O "nome" pode ter no máximo 50 caracteres, a "idade" deve ser um número inteiro e o "telefone" pode ser um vetor de no máximo 20 caracteres.

Implemente as opções de “Inserir”, “Alterar”, “Excluir” e “Exibir” contatos. Para as opções de “Alterar”, “Excluir” e “Exibir”, o programa deve perguntar ao usuário o nome do contato e, em seguida, realizar a respectiva operação.

Obs.:

A) Não utilize o “struct” da linguagem C.

B) Uma opção de implementação seria:

1) no início do programa, todo o conteúdo do arquivo é carregado para vetores na memória.

2) durante a execução do programa, os dados são manipulados por meio dos vetores na memória.

3) Ao final do programa, os vetores são gravados no arquivo.

Você NÃO deve implementar o programa desta maneira. Todas as operações (“Inserir”, “Alterar”, “Excluir” e “Exibir”) devem ser feitas diretamente no arquivo. Por exemplo, quando o usuário escolher a opção “Alterar”, o programa deve encontrar o contato diretamente no arquivo e alterar seu conteúdo.

C) Veja, na página 231 do livro texto, os modos de abertura de arquivo. Possivelmente, será necessário usar o modo “a+b”

Estruturas (Capítulo 11)

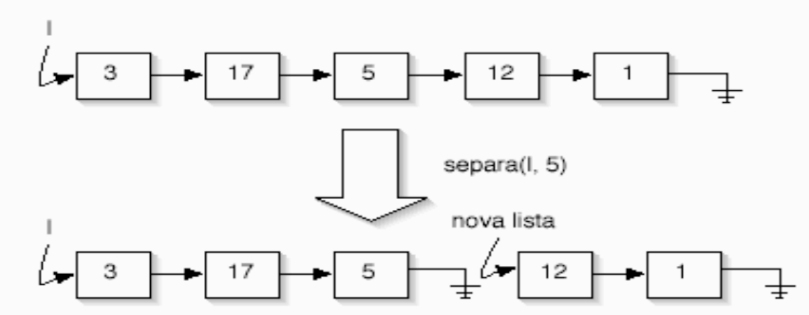
89) Implemente o exercício anterior usando estruturas.

90) Implemente um vetor de estruturas onde cada elemento é constituído por um Número positivo e por um Nome (40 caracteres). O vetor não poderá ter mais do que MAX registros (elementos). Implemente as seguintes funções:
    1. Inserir, que adiciona um registro ao vetor e retorna a posição em que foi introduzido.
    2. Salvar, que salva em arquivo binário o conteúdo do vetor.
    3. Carregar, que faz a leitura dos registros existentes no arquivo e os insere no vetor. Se o registro já existir no vetor, ele não deverá ser inserido.
    4. Listar todos, que lista todos os elementos do vetor.

Remover registro, que solicita um Número e remove do vetor o registro correspondente.

Memória Dinâmica (Capítulo 12)

91) Considere uma lista encadeada simples de valores inteiros, como mostra a figura.



Implemente a função “*separa*” que recebe como parâmetro a lista encadeada e um valor inteiro *n* e divide a lista em duas, de tal forma que a segunda lista comece no primeiro nó logo após a primeira ocorrência de *n* na lista original. A figura acima ilustra esta separação.

A função deve obedecer ao protótipo: *Lista* separa (Lista* l, int n)*; A função deve retornar um ponteiro para a segunda sub-divisão da lista original, e o parâmetro "*l*" deve continuar apontando para o primeiro elemento da primeira subdivisão da lista. Implemente o programa principal para testar sua função. A lista deve ser alocada dinâmicamente e preenchida no programa principal.

92)

Considere estruturas de listas encadeadas que armazenam valores inteiros. O tipo que representa um nó da lista é dado por:

```
struct lista {
   int info;
   struct lista* prox;
};
typedef struct lista Lista;
```

Implemente uma função que receba um vetor de valores inteiros com `n` elementos e construa uma lista encadeada armazenando os elementos do vetor nos nós da lista. Assim, se for recebido o vetor `v[5] = {3, 8, 1, 7, 2}`, a função deve retornar uma nova lista cujo primeiro nó tem a informação `3`, o segundo a informação `8`, e assim por diante. Se o vetor tiver zero elementos, a função deve ter como valor de retorno uma lista vazia. O protótipo da função é dado por:

```
Lista* constroi (int n, int* v);
```

93) Implemente uma agenda de contatos que armazena o "nome", "celular" e "telefone". Salve os dados da agenda em arquivo binário. Durante a execução do programa, os dados da agenda deverão ser manipulados utilizando-se uma lista dinâmica. Ao iniciar o programa, carregue os dados do arquivo para a lista dinâmica e, ao fechar o programa, salve as informações da lista no arquivo binário. Implemente as funções de "incluir", "excluir" e "procurar" um contato.